salesforce

Multi-Signer DNSSEC

DNSSEC Provisioning Panel
ICANN 68

Shumon Huque
June 22nd 2020

DNSSEC with Multiple Providers

- **Zone Transfer Model**
  - **Zone Owner signs the DNS zone** on the hidden master they run, pushed out the signed zone via zone transfer to the providers.
  - Works fine.
  - Doesn't support dynamic, traffic management features that require the providers to perform signing.
- **Provider API Model**
  - Zone Owner uses Provider APIs to update zone content identically at each provider.
  - **Supports dynamic, traffic management features.**
  - **Each provider has their own DNSSEC keys** that they sign the zone with.
  - This requires new key management and co-ordination protocols.

Today’s Focus

DNSSEC with Multiple Providers

- **Zone Transfer Model**
  - **Zone Owner signs the DNS zone** on the hidden master they run, pushed out the signed zone via zone transfer to the providers.
  - Works fine.
  - Doesn’t support dynamic, traffic management features that require the providers to perform signing.
- **Provider API Model**
  - Zone Owner uses Provider APIs to update zone content identically at each provider.
  - **Supports dynamic, traffic management features.**
  - **Each provider has their own DNSSEC keys** that they sign the zone with.
  - This requires new key management and co-ordination protocols.

Dynamic response mechanisms

- Often called "**Traffic Management**"
  - Global Server Load Balancing (GSLB), Probe & Failover records, Weighted response, Custom programmed dynamic responses, etc.
  - Non-standardized, hence incompatible with the DNS zone transfer protocol.

- Often querier-specific or dependent on inspecting other dynamic state in the network
  - So answer and signature typically have to be determined at the authoritative server answering the query, at the time of the query, or both.
  - **This necessarily means the DNS provider must be able to sign with their own DNSSEC keys.**

Multi-Signer Models

Multi-Signer DNSSEC models

- Each DNS provider signs zone data with their own keys.
- Zone owner uses provider specific zone management APIs to update zone content at each provider.
- A set of new key management mechanisms have been developed to make this model work:
- https://tools.ietf.org/html/draft-ietf-dnsop-multi-provider-dnssec-05
- Currently in the RFC Editor queue – should be published as an RFC in the near future.

Multi-Signer DNSSEC models

- Support the non-standard DNS features *if* the provider is capable of signing the response data generated by these features.

- Common strategies for doing so:
  - On-the-fly signing (Online signing)
  - Pre-compute & sign all possible response RRsets, and then algorithmically determine at query time, which response + signature to return.

Key Management Requirements

- Main requirement: manage the contents of the DNSKEY and DS RRsets such that validation is always possible, no matter which provider you query and obtain the response from.

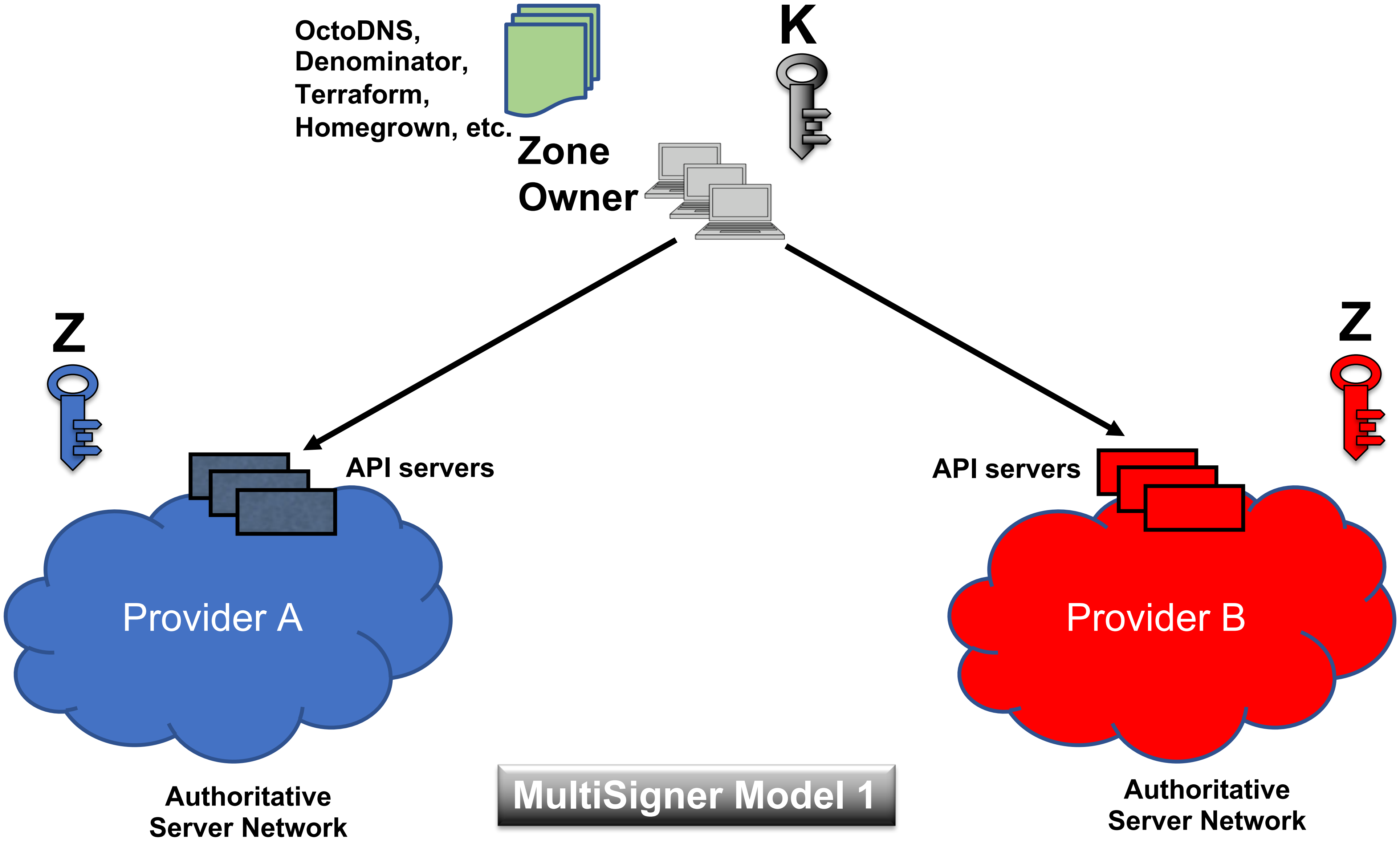
- **Strategy: each provider has to import the zone signing (public) keys of the other providers into their DNSKEY RRset.**

- (See section 3 of the protocol draft for detailed rationale)

Multi-Signer Model 1

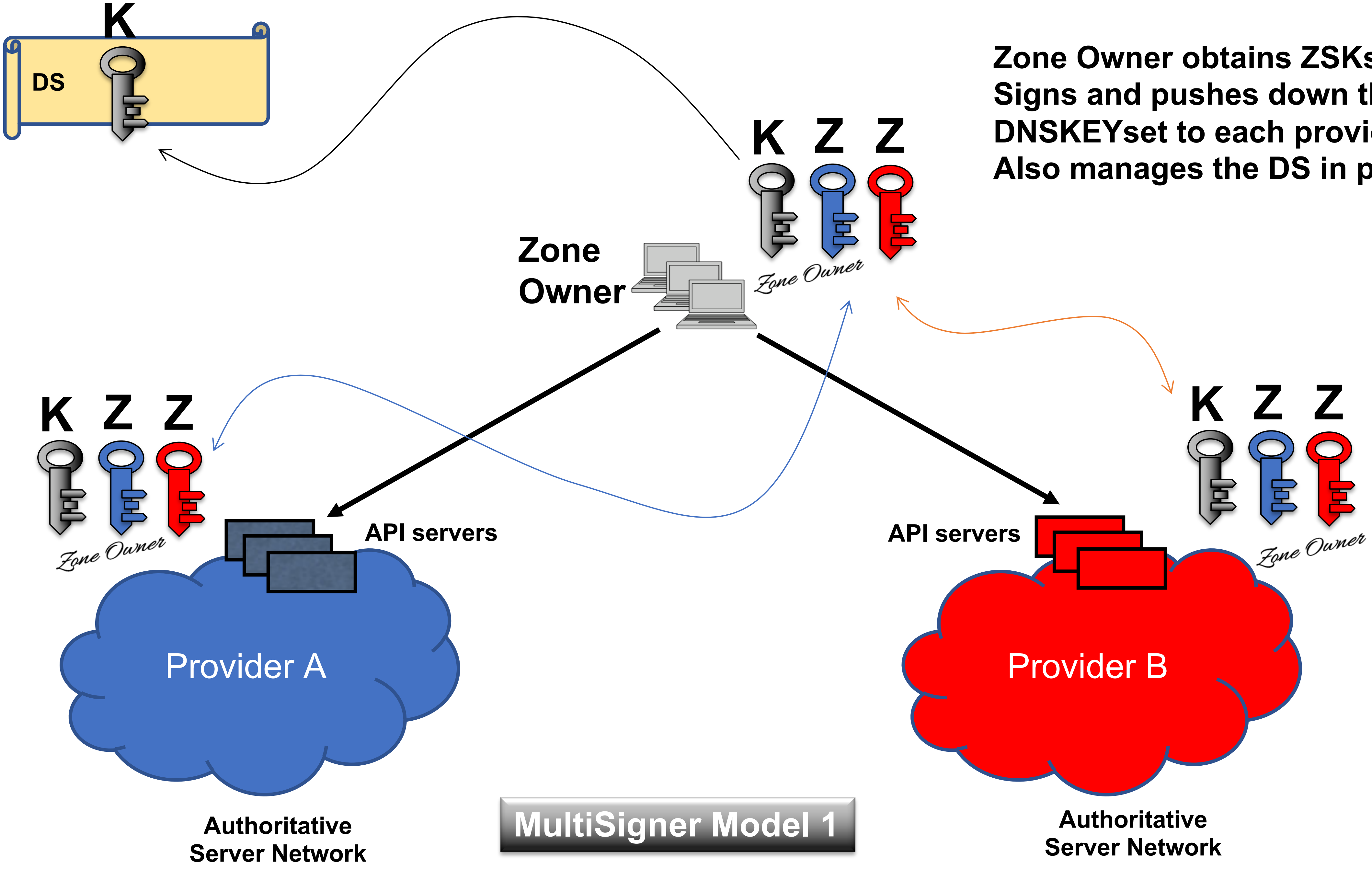
- Common KSK; Unique ZSK per Provider

Zone Owner controls KSK
Each provider has a ZSK



MultiSigner Model 1

Zone Owner obtains ZSKs, Signs and pushes down the DNSKEYset to each provider. Also manages the DS in parent.



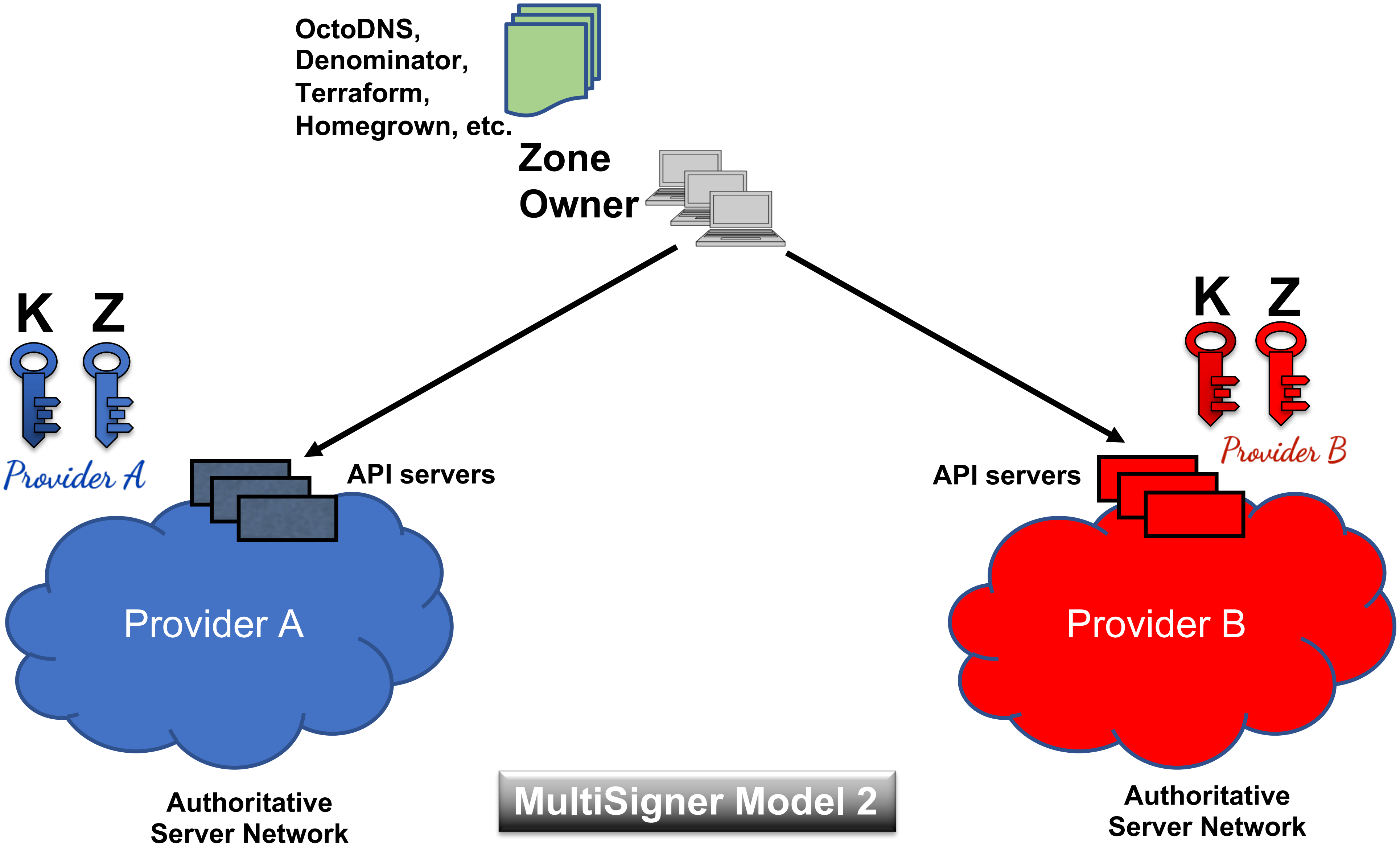
MultiSigner Model 1

Multi-Signer Model 2

- Unique KSK & ZSK per Provider

MultiSigner Model 2

Each provider has their own KSK and ZSK.



OctoDNS,
Denominator,
Terraform,
Homegrown, etc.

Zone Owner

K Z

Provider A

API servers

Provider A

Authoritative Server Network

K Z

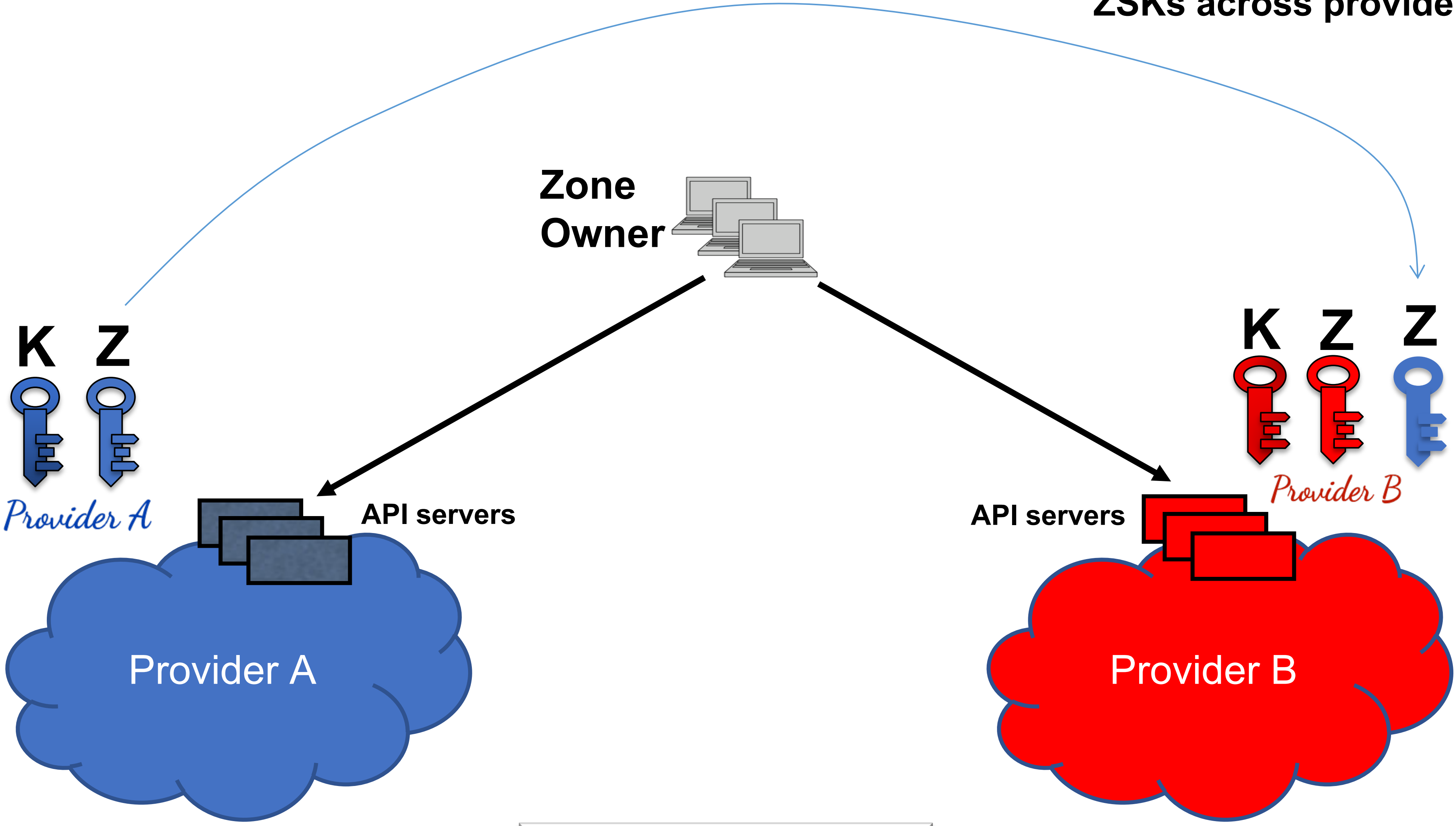
Provider B

API servers

Provider B

Authoritative Server Network

We need to cross-import ZSKs across providers.



Zone Owner

K Z

Provider A

API servers

Provider A

Authoritative Server Network

K Z Z

Provider B

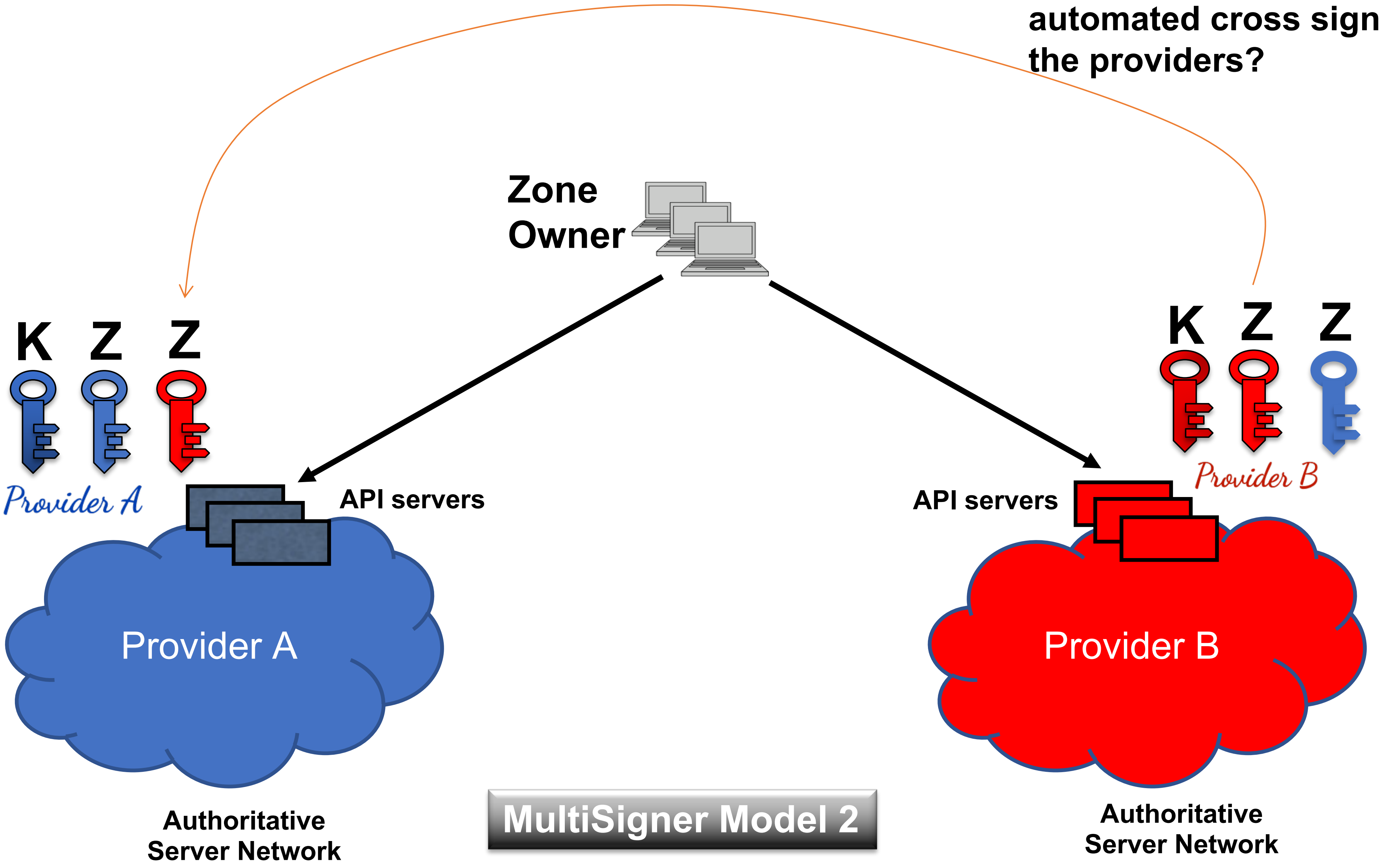
API servers

Provider B

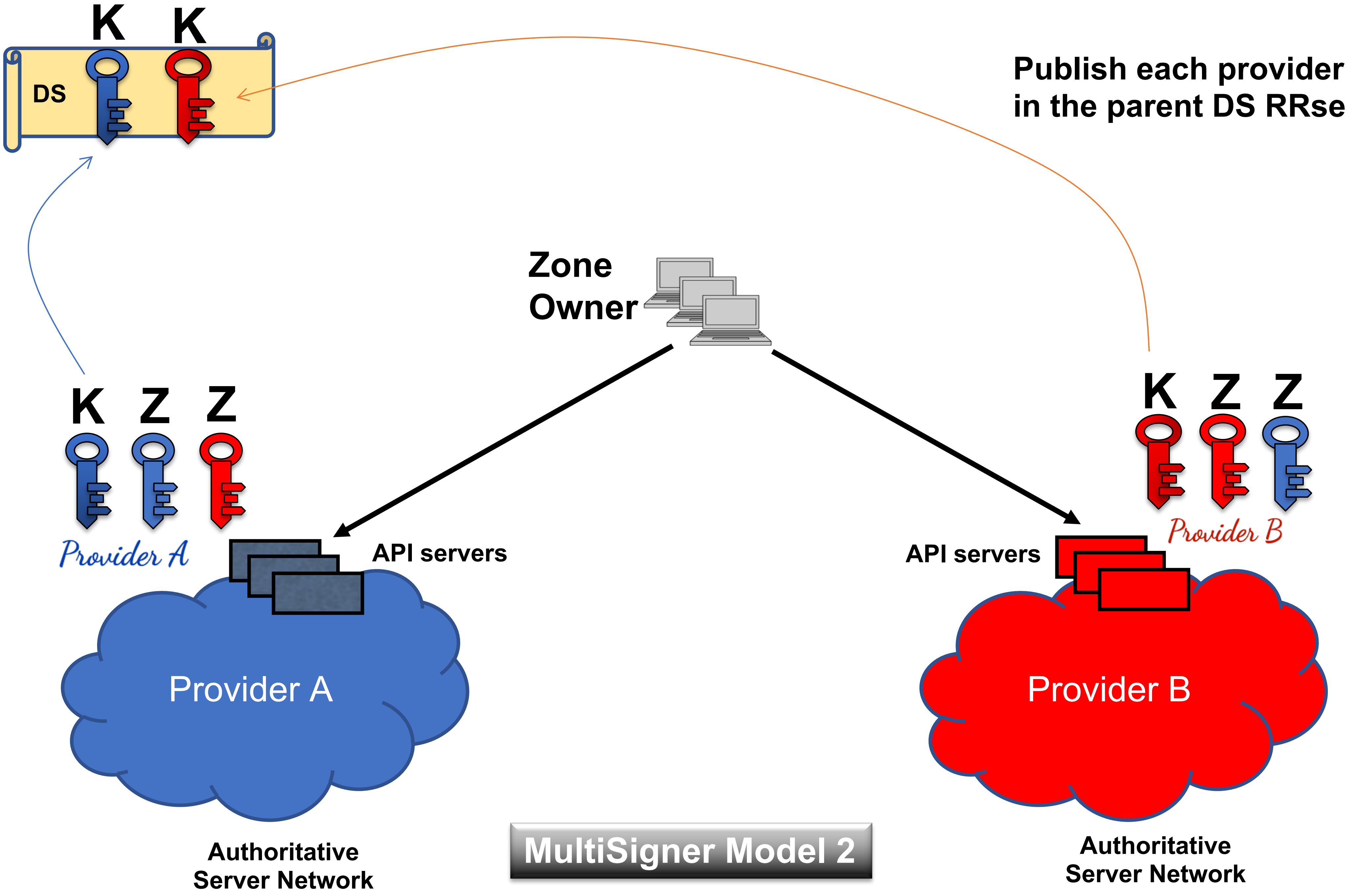
Authoritative Server Network

MultiSigner Model 2

Should we devise a protocol for automated cross signing between the providers?



MultiSigner Model 2



Publish each provider's KSK
in the parent DS RRset.

MultiSigner Model 2

Software Toolkits & Aids

Multi-Provider Software Toolkits

- Software toolkits to help keep zone contents consistent across the multiple providers are usually important.

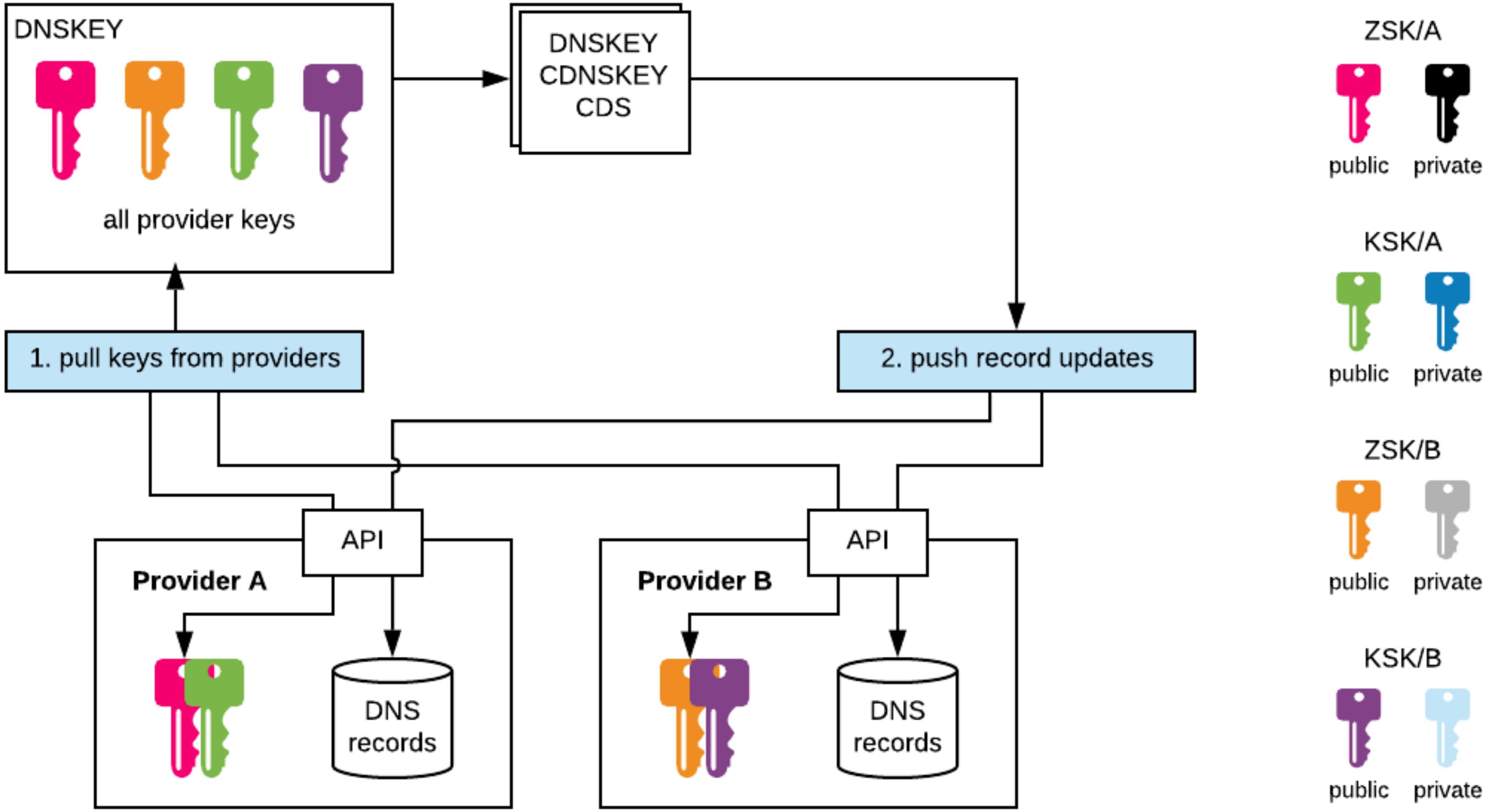
- Existing opensource ones in common use:
  - OctoDNS
  - Denominator
  - Terraform, and others

Implementation & Deployment Status

Implementation & Deployment

- Protocol still quite new
- But has been tested in hackathons and lab environments
- NS1 has a production implementation:
  - https://ns1.com/press/ns1-salesforce-collaborate-on-multi-signer-dnssec-implementation
- Various open source prototypes exist
- At least 2 other DNS vendors are working on implementations

NS1 API for Model 2



DNSKEY

all provider keys

DNSKEY
CDNSKEY
CDS

1. pull keys from providers

2. push record updates

API

Provider A

DNS records

API

Provider B

DNS records

ZSK/A

public private

KSK/A

public private

ZSK/B

public private

KSK/B

public private

DS Automation Tie-in

DS Automation Tie-in

- Model 1:
  - CDS/CDNSKEY works fine
  - Extending OctoDNS, Terraform etc to talk to Registrars
    - Hexonet is a SaaS backend for several hundred registrars (from Jothan)
    - Plugin to talk to it and other key registrar systems could be useful.

- Model 2:
  - CDNS/CDNSKEY works (but coordination needed)
  - Registrar mediated protocols (DomainConnect?)
  - Delegated authorization for Operators? (DS only)
  - Formally designating (multiple) operators in the RRR system?

thank you
BLAZE YOUR TRAIL
salesforce